

- 1 22. (currently amended) A data processing system having software stored in a set of
2 Computer Software Storage Media for translating blocked data transferred from a
3 program executing on one of a plurality of computer systems to another of the
4 plurality of computer systems, wherein:
5 the plurality of computer systems comprises:
6 a first computer system containing a first program communicating through
7 an API with a first interface system, and
8 a second computer system containing a second interface system for
9 communicating with the first interface system;
10 the first computer system and the second computer system are heterogeneous
11 computer systems coupled together over a communications link;
12 said software comprising:
13 A) means for opening a first session from the first program via the API through
14 the first interface system to the second interface system;
15 B) means for specifying a first translation for records transmitted over the first
16 session;
17 C) means for blocking a first plurality of records into a first block of records;
18 D) means for transmitting the first block of records over the first session from a
19 ~~the first one of the plurality of~~ computer systems to a the second ~~one of~~
20 ~~the plurality of~~ computer systems ;
21 E) means for unblocking the first block of records into the first plurality of
22 records on the second ~~one of the plurality of~~ computer systems; and
23 F) means for translating each of the first plurality of records in accordance with
24 the translation specified in set (B).

REMARKS

This amendment is presented in response to the Office Action mailed on September 22, 2004 for the purpose of placing the application for reconsideration and allowance. Claims 1-22 are active in the application.

Applicants have made minor changes to the specification for the purpose of updating the status of a related patent application which has issued into a patent and for the purpose of correcting typographical errors. Also, Applicants have received an Office Action in another one of the related patent applications. For the convenience of the

Examiner, a copy of the list of prior art patents in Office Action is included as an attachment herein. Applicants have also made minor amendments to the claims. The amendments to the claims are editorial in nature and were made for the purpose of consistency in the antecedents contained in the claims. The amendments were not made in any way for the purpose of distinguishing over the prior art.

Claim Rejection under 35 U.S.C. 102 (b)

Applicants traverse the Examiner's rejection of claims 1-3, 5,6, 8-13, 15, 16 and 18-22 20 under 35 U.S.C. 102(b) as being anticipated by U.S Patent No. 5,926,636 to Lam (hereinafter Lam).

The Examiner Has Not Established a Prima Facie Case of Anticipation

The Examiner is required to show that each and every element as set forth in the claim is found in the Lam patent. Also, the Examiner must show that the identical invention is shown in as complete detail as contained in the claim and that the elements must be arranged as required by the claim.

Applicants traverse the Examiner statement that Lam teaches the invention as claimed including methods for managing components in a heterogeneous computer system network. By contrast, Applicants claims are directed to translating blocked data transferred from a program executing on one of a plurality of computer systems to another of the plurality of computer systems.

As noted by the Examiner, Lam is directed to a remote procedural call (RPC) component management method for a heterogeneous computer network. In response to a component management function call by a remote client application, the component management application programming interface (API) generates a message that identifies the called function and the version of the component management API (see abstract). The "Background of the Invention" portion of the Lam patent notes that one problem encountered using RPCs is the **representation of data** across a network with multiple platforms because different CPUs represent data structures differently (for example, Big-Endian versus Little-Endian). Lam addresses this problem as described in connection with Figure 4. Lam in discussing Figure 4 describes that the RPC command client **converts the function call and any associated data and any associated data to a neutral canonical format**. This conversion approach is also discussed in the text

entitled "Modern Operating Systems" by Andrew S. Tanenbaum, copyright 1992, Prentice-Hall, Inc. at pages 424-425.

It should be also noted that Lam distinguishes RPC as a departure from DMI which uses data blocks to describe the format for data transfer instead of parameters to a function call. Thus, Lam views DMI as being data oriented while RPC is procedural and message oriented. Lam is directed to using RPC for supporting different versions of application programming interfaces by remote procedure call modules on client and server computers.

Claim 1

Accordingly, in view of the above and relative to Claim 1, Applicants do not find Lam to teach a method of translating blocked data transferred from a program as stated by the Examiner. In the Lam system, in response to a remote client component management function call, the remote client component management API builds a message in a buffer memory of the remote client computer. Thus, there is no blocking of a plurality of records into blocks of records as specified in Claim 1.

Lam does teach the use of a field or identifier for a version of the component management API for the purpose of determining whether the **addressing format** of the client computer is compatible with an addressing format of the server computer. For example, an addressing format field could be coded for indicating whether addressing is express or implied, using 32 or 64 bit addressing. Further, Lam distinguishes addressing format (architecture) from data structures.

The above distinction is discussed in column 12, lines 26-36 relative to the server's processing of an RPC message. More specifically, Lam explains that its server component management function module parses the message RPC Message Request to determine the client machine architecture specified in the message. The message is stated as being in the format shown in Appendix A (**missing from the patent**) and contains a byte order field used to indicate **the computer architecture** of the remote client computer. The information in byte order field is independent of the order that the byte order field is processed. Consequently, the byte order field is read correctly independent of **the addressing method** used by a **particular computer architecture**. In

implementing this, Lam uses the step of aligning information in a message buffer so that the information is one byte-aligned (see in Claim 2 of the patent).

In view of the above and upon review of the material cited by the Examiner, Applicants find Lam absent opening a first session from the first program via the API through the first interface system to the second interface system. The material cited by the Examiner referenced in column 5, lines 3-11, and lines 16-20 describes a client program filling a buffer by calling a function in the API (making a buffer transfer function call), issuing a message transfer RPC and performing a version comparison. Applicants find no mention in the cited material of opening a session via the client program.

Applicants find Lam absent specifying a first translation for records transmitted over the first session. Lam as noted by the Examiner is concerned with messages (e.g. `RPC_MESSAGE_REQUEST`) and **not records**. It will be noted that the specification of the subject application also discusses messages but the subject claims are not directed to such subject matter.

The material cited by the Examiner in column 5, lines 35-42 discuss the use of an identifier which defines **the computer architecture** (i.e. addressing format) and the conversion of a message to a format that is compatible with the second computer (i.e. addressing format). As discussed in column 12 of Lam, when the message `RPC_MESSAGE_REQUEST` is in a format that is different from that processed by I/O manager server 631, server component management function module 623 converts the format of the message `RPC_MESSAGE_REQUEST` to a **neutral canonical format** message `IPC_MESSAGE` using an external data representation (XDR).

From the above, it is seen that Lam is concerned with converting the addressing formats of messages and not the conversion of data or record structures. Accordingly, Applicants submit that Lam does not disclose blocking a first plurality of records into a first block of records. The material cited by the Examiner in column 6, lines 1-4 discloses packaging messages for transfer over the computer network. The packaged message is transmitted to a first network stack on the client computer which in turn transmits the packaged message from the first network stack to a second network stack on a server computer. From this, it should be clear that the process of packing is not

equivalent to the process of blocking. As known in the art, the idea behind RPC is to make a remote procedure call look as much as possible like a local one. For example, when a read is actually a remote procedure (one that will run on the file server's machine), a different version of read, called a client stub is put into the RPC function library. Instead of the normal calling sequence that puts the parameters in registers and ask the kernel to give it data, it packs or builds the parameters into a message and asks the kernel to send the message to the server. When the message arrives at the server, the kernel passes it up to a server stub that is bound with the actual server. The server stub unpacks the parameters from the message and then calls the server procedure in the usual way. To conclude that the process of packing and the process of blocking are equivalent would be like concluding that the use of instructions and data are the same. As discussed above, RPC interface is a procedural interface.

Accordingly, Applicants submit that Lam does not disclose transmitting the first block of records over the first session from a first computer to a second computer for the reasons discussed above. For similar reasons, Lam does not disclose unblocking the first block of records into the first plurality of records on the second computer. As noted by the Examiner, the material cited in column 6, lines 6-8 discloses packed messages converted back to the message. As indicated above, the process of converting messages which is a procedure oriented stack operation involving parameters of a message should not be deemed equivalent to that of unblocking data records.

As discussed above, Lam does not disclose translating each of the first plurality of records in accordance with the translation specified in step (B). The cited material refers to an identifier of the computer architecture (addressing format) and if the computer architecture of the first computer is incompatible with the second computer, the message is converted to a form that is compatible with the second computer. Thus, Applicants submit that there is no translation of records let alone a translation of records as specified in an earlier step as specified in claim 1. The absence of the above elements and functions should be persuasive that Lam does not anticipate Applicants claims. A notice to this effect is respectfully solicited.

Claim 2

For the reasons given above relative to claim 1, Lam is absent a teaching of performing a translation step by a first interface as specified in claim 2. Lam is absent a translation step in that the conversion of a message format should not be deemed the equivalent of the specified step of claim 2.

Claim 3

For the reasons given above relative to claim 1, Lam is absent a teaching of performing a translation step by a second interface. The material cited in column 6, lines 12-16 cited by the Examiner pertains to conversion of an **addressing format** of a **message** compatible with the server component. By contrast, claim 3 is directed to translation of records.

Claim 5

As discussed above, Lam is not concerned with records let alone records, each of which comprise a plurality of fields and the translation of an integer format from a first format to a second format. The material cited by the Examiner in column 12, lines 44-49 concern the translation of a message request that is in an address format that is different from that processed by the I/O manager server 631. Lam states that the server component converts the format to a neutral canonical format message. Lam provides an example of this which corresponds to the material cited by Examiner-that of converting a message in a little endian **addressing** format to a big endian **addressing** format. Again, Lam is concerned with addressing formats and not data record representations. Also, Lam is not concerned with translating an integer in one of the plurality fields in a record from a first integer format to a second integer format.

Claim 6

For reasons given above, Applicants find Lam absent translating an integer from a first endian format to a second endian format as specified in claim 6. Again, Lam is concerned with addressing formats and not data formats as defined in claim 6.

Claim 8

Applicants find Lam absent step B, let alone disclosing the use of a file containing a record description as recited in claim 8. The material cited by the Examiner contained in column 5, lines 35-42 pertains to a field read in a message that includes an identifier of the computer architecture of the first computer as discussed above. Applicants submit

that the use of an identifier specifying the addressing format of a computer should not be deemed the equivalent of using a file containing a record description for specifying translation of record fields within a record according to claim 8.

Claim 9

For reasons given with respect to claim 8, Applicants find Lam absent the specifying step (B) let alone utilizing a memory area containing a record description as specified in claim 9. The material cited by the Examiner at column 5, lines 47-53 pertains to the building/packing of the message in a buffer memory. It does not discuss including the identifier in such buffer memory. In fact, the identifier is used to determine if the message addressing format must be converted and therefore would not be included. This arrangement is contrary to that recited in claim 9.

Claims 10-13, 15, 16 and 18-22

For the reasons given above, claims 10-13, 15, 16 and 18-22 should also be deemed patentable over the cited teachings of Lam. Further, claims 10 and 20 specify the opening of a second session from the first program. For the reasons discussed above relative to claim 1, Applicants submit that Lam does not teach the opening of a first session let alone the steps specified in claims 10 and 20.

Claim Rejection - 35 U.S.C. § 103

Applicants submit that it is impermissible within the framework of section 103 to pick and choose from any one reference only so much of it as will support a given position, to the exclusion of other parts necessary to the full appreciation of what such reference fairly suggests to one of ordinary skill (see 147 USPQ at 393). Applicants traverse the Examiner's rejection of claims 4, 7, 14 and 17 under 35 U.S.C. 103(a) as being unpatentable over Lam further in view of Allen, U.S. patent no. 6,658,625 (hereinafter Allen).

The basis of the rejection is that (1) Lam fails to teach translating a first character format to a second character format and translating a first floating point format to a second floating point format and (2) Allen teaches a generic data converter that uses a data description to convert data along with the use of a floating point converted to another floating point and the use of converting character sets. According to the

Examiner, it would have been obvious to one of ordinary skill in the art to modify Lam in view of Allen to translate a first character format to a second character format and translate a first floating point format to a second floating point format. The motivation cited by the Examiner is that one would be motivated to do so because it would allow for translation of different types of data.

First of all, the material cited in Lam by the Examiner does not pertain to the data formats or types but rather addressing formats of computers relative to transfer of messages between computers. Lam provides conversion of messages into a neutral canonical format message. Once such conversion of messages takes place, the messages can be processed by the receiving server system. Other than the Examiner's statement, Applicants submit that there is no reason to provide additional conversions or different conversions to accomplish the objectives of the Lam patent. The material cited by the Examiner in Allen pertains to document processing and expediting the parsing of the data elements contained in such documents by a generic data converter through the use of a data description (XML parser output that has been placed into a hash table object).

Neither Lam nor Allen teach the use of data records, let alone records that comprise a plurality of fields, one of which is an alphanumeric field and the translation of each character in one such field from a first character format to a second character format as defined in claim 4. Similarly, relative to claim 7, neither Lam nor Allen teach the use of data records, let alone records that comprise a plurality of fields, one of which is a floating point field and the translation of each floating point numbers in that one field from a first floating point format to a second floating point format. The material cited by the Examiner in column 16 discusses the converting of character sets through the use of additional description to a description of hostName and the use of a code page that allows the data converter to know what language the characters are in the data and what and how to convert them to the equivalent language in Unicode. Clearly, since Lam and Allen are directed to different processing schemes, Applicants submit that there would be no motivation to try to combine such teachings, let alone modify Lam in the manner specified by the Examiner.

For similar reasons, claims 14 and 17 should also be deemed patentable over the proposed combination of Lam and Allen. Further, claim 17 specifies carrying out such

translation through the use of a set of computer instructions while Allen teaches the use of a generic data converter.

In view of the above arguments and clarifying amendments, Applicants submit that claims 1-22 should be deemed patentable over the cited prior art. A notice to this effect is respectfully solicited.

Applicants ask the Examiner to contact Applicants attorney to discuss any other grounds for rejecting Applicants claims before acting on this amendment.

Also, if any questions or issues should arise with respect to this amendment or the allowability of this application, the Examiner is **urged to call Applicants' attorney at the number indicated herein**. Further, if the Examiner feels that a discussion will further advance the prosecution of this application, the Examiner is also **urged** to call as suggested herein.

Respectfully submitted,

A handwritten signature in cursive script that reads "Faith F. Driscoll".

Faith F. Driscoll
Registration No. 24,206
Attorney for Applicants
(781) 326-6645

Encl: Attachment
FFD/fd